Packet Radio Station Hardware, Operating System and Applications Programming Environment+

J.Burchfiel, R.Tomlinson, M.Beeler*

Packet Radio Temporary Note 138

April 15, 1975

****

+This research was supported in part by the Advanced Research Projects Agency of the Department of Defense under Contract No. DAHC15-71-C-0088.

*J.Burchfiel, R.Tomlinson and M.Beeler are with the Computer Science Division of Bolt Beranek and Newman, Cambridge, Mass.

I. Introduction

A packet radio network is a digital broadcast channel, fixed and mobile digital terminals which are sources and sinks of information, stations which provide centralized routing control and interconnections to other networks, and repeaters which provide area coverage for mobile terminals by performing a store-and-forward function on the radio broadcast channel. An experimental testbed for these concepts is now under construction at the Stanford Research Institute, Palo Alto, California to provide experimental validation of theoretical models and simulation predictions [Ref. 1,2,3,4,5,6].

Objectives of this technology exploration include use of a broadcast band in digital burst mode to support digital computer and terminal communication, demonstration of coexistence with existing broadcast applications, and secure mobile communications through encryption techniques.

The Packet Radio Network (PRN) concept was initially explored in the ALOHA Project at the University of Hawaii. (Ref. 2) Such a network is different from point-to-point packet switched networks in a number of significant areas:

1. The use of mobile terminals requires facilities for tracking and handoff of the terminals from repeater to repeater and station to station.

2. The use of a common broadcast channel results in collisions due to simultaneous packet transmissions: the communications paths require extensive error checking and correction to handle the resulting high error rate.

3. The repeater is required to operate unattended in relatively remote areas for long periods of time. It must therefore be a simple, low-powered component. Accordingly, it is not designed to support complex dynamic routing algorithms. Instead, such functions must be provided in the stations, giving some measure of centralized control in the Packet Radio Network.

On the other hand, a PRN has many features and functions in common with a point-to-point packet switched network:

1. The PRN must provide routing which adapts dynamically to component failures.

2. the PRN must support interprocess connections which have flow control and error control.

3. The PRN must have centralized monitoring, debugging, and statistics collection tools to provide maintenance and performance evaluation capabilities.

The dynamic packet routing capability (packet store and forward) is programmed in the repeaters, with the stations providing initialization and centralized control of parameters for terminal tracking. The programmable capability of the repeater is provided in its IMP-16 microprocessor.

Reliable data transmission between PRN data sources and sinks is required in spite of errors and transmission 'collisions' on the broadcast channel. This is achieved by defining a logical entity called a 'connection' between the source process and destination process, and performing end-to-end error detection and correction over this noisy channel. The connection is thus a reliable (error corrected) data transmission path.

The programs for providing interprocess connections within the PRN must be programmed into each terminal (data connection), each repeater (control connection), and each station (data and control connections.)

In addition to the communications support, the terminal must also have a terminal handler program which manages terminal input and output buffers and performs translation of format effector characters as needed.

The programs which provide centralized monitoring, debugging, and statistics collection are located in the station, with a small (slave) routine in each repeater. These functions are shown in Figure 1.

Digitized by the Internet Archive
in 2025 with funding from
Amateur Radio Digital Communications, Grant 151

https://archive.org/details/temporary-note-138

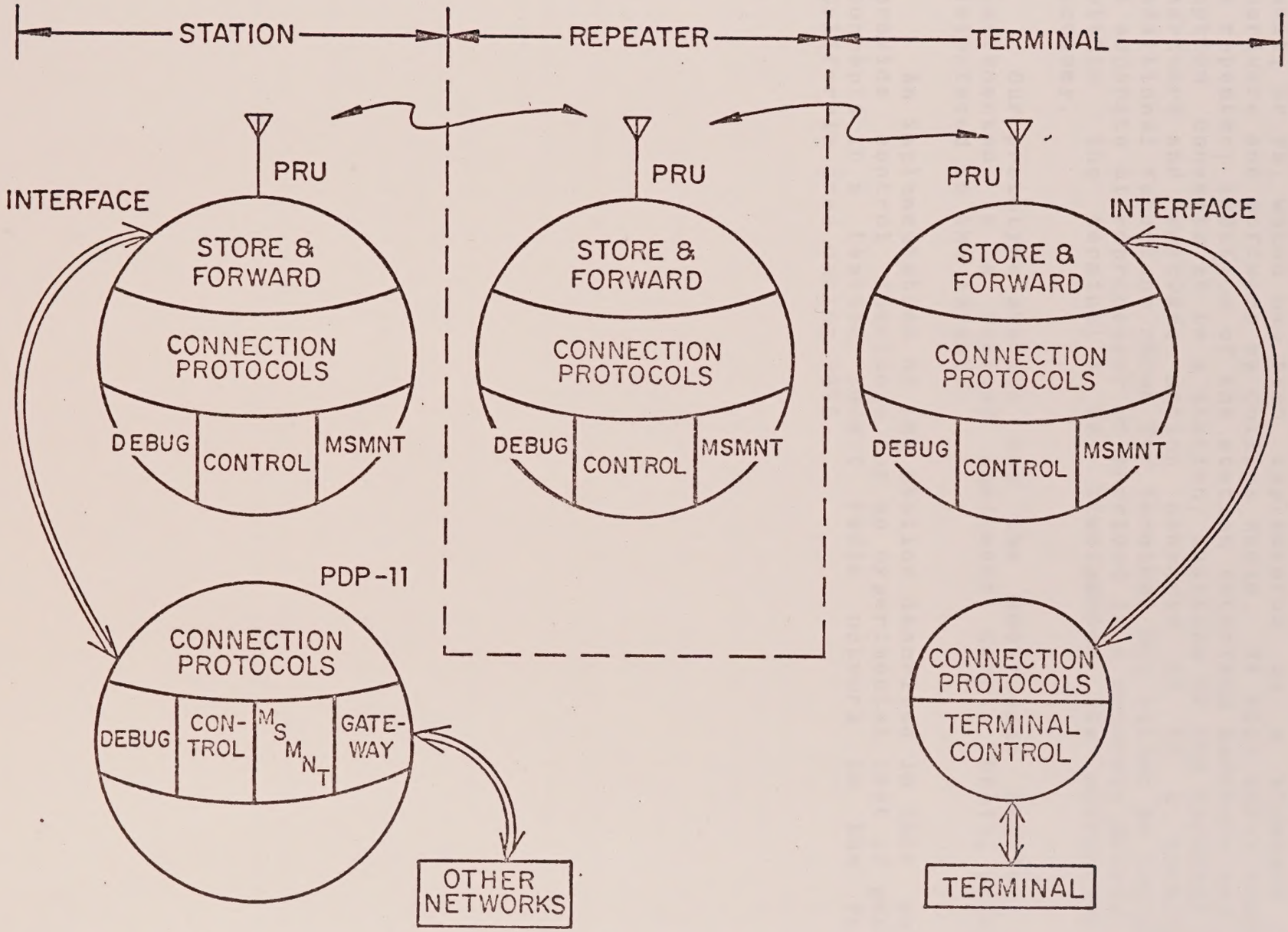


FIGURE 1: FUNCTIONS OF A PACKET RADIO STATION, REPEATER AND TERMINAL

The set of functions which appear in common in a station, repeater, and terminal are identified in Fig. 1 as a Packet Radio unit, or PR, which has been implemented as a standard piece of hardware and software by Collins Radio. It will serve standalone as a repeater; addition of the station interface hardware and software option converts it to a station; addition of the terminal interface hardware and software option converts it to a terminal. The additional functions shown for terminal may either be implemented in a separate microprocessor or provided in a separate memory partition within the terminal's PR, timesharing its microprocessor for economy.

Our prototype station has the additional station functions implemented in a Digital Equipment Corp. PDP-11, which is also interfaced to the ARPANET.

An implementation of the station described in this paper will provide control functions for an experimental test of packet radio concepts in a testbed packet radio network in the Palo Alto, California area during 1975.

II. Overview of Station Functions

The Stations of a packet radio network are locations of centralized control. They are responsible for direct control of routing within the network: this involves an initialization procedure which measures the propagation connectivity between the packet radio units (PR's) which are the active radio sections of stations, repeaters and terminals. This procedure also establishes a control connection from a station to each PR it controls to initialize its routing parameters. The control functions of a Station also include dynamic updating of PR routing parameters to route traffic around repeaters which fail. Since many PRN terminals are mobile, this dynamic updating of routing parameters must handle automatic tracking of terminals and hand-off from repeater to repeater and from station to station as a terminal moves through the network.

A second function supported by a Station is centralized debugging: this permits examination and modification of the memory contents of any PR from a Station, as well as the collection of reports of exception or error conditions anywhere in the network. A Station also serves as a central collection and reporting point for all PRN traffic statistics and performance measurements.

A third function of a Station is to support a standard TELNET protocol which makes unencrypted terminals attached to the PRN appear as network virtual terminals which are compatible with the ARPANET and which can access the services and resources of ARPANET hosts. Encrypted PRN terminals (using end-to-end encryption) must provide enough local intelligence to support the TELNET protocol for themseles.

A Station supports internetwork communication between host computers on the PRN and hosts on the ARPANET by performing the internet gateway functions of conversion between local network protocols. This requires address mapping, flow control and error control. It is also responsible for accounting and directory functions.

The station will also be responsible for allocating packet radio network resources to prevent congestion on the broadcast channel.

III. Station Control of Connections in the Packet Radio Network

The basic facility required in the PRN is support of interprocess connections which provide reliable delivery of data from a PRN source to a PRN destination. Such connections require flow control to prevent a source from overloading the network and causing serious congestion. They also require error control because of message interference in the shared broadcast communication channel. The error control mechanisms selected are:

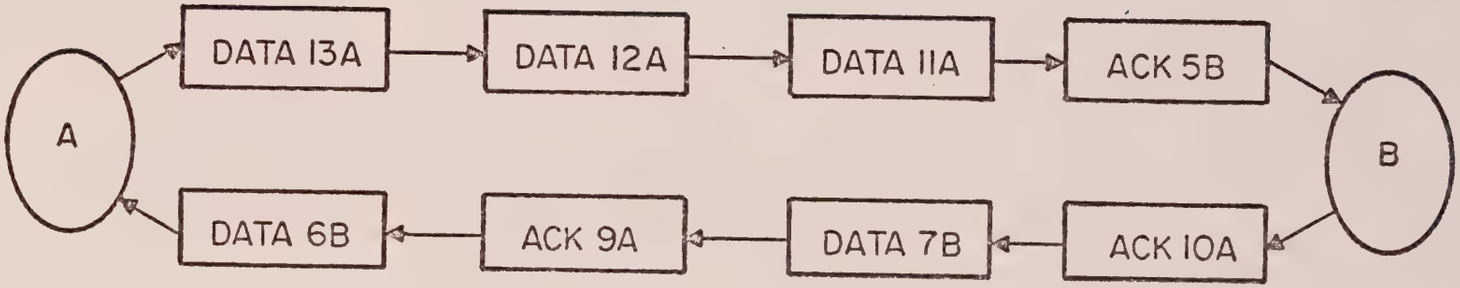
1. A sequence number in each packet to permit detection of missing or duplicate packets.

2. An end-to-end positive acknowledgement for packets which arrive successfully.

3. A source timeout which causes periodic retransmissions of unacknowledged packets.

Since most expected uses of the PRN will require bi-directional communication, the PRN connection is bi-directional, with flow control and error control information for data transmission in each direction piggybacked onto the data flow in the opposite direction. This arrangement is depicted in Figure 2.

Certain applications of packet radio will not require reliable delivery of data, (e.g. real time seismic or speech data) and such devices will not be required to support the connection protocol described above.



A STATUS:

SENT	RECEIVED
DATA 13A	ACK 8A
ACK 5B	DATA 5B

B STATUS:

SENT	RECEIVED
DATA 7B	ACK 4B
ACK 10A	DATA 10A

FIGURE 2: FULL DUPLEX INTERPROCESS CONNECTION

Figure 2 shows that multiple packets of data can be in transit in both directions at one time. The acknowledgement are cumulative, i.e., an acknowledgement carrying serial number 9 acknowledges successful receipt of all packets up to and including packet number 9. It is therefore unnecessary to send out an acknowledgement for each data packet received: one 'ack' can cover a number of data packets. Further, the 'ack' carries such a small amount of information, that it can be inserted into the header of a data packet travelling in the same direction, resulting in very low overhead for this acknowledgement mechanism. However, timely acknowledgement is required to prevent source timeout and retransmission, so if there is no data traffic which can be used in this piggyback fashion for 100 milliseconds or so, a separate 'ack' packet must be generated and sent.

The status information which must be maintained at each end of the connection is minimal: four sequence numbers. The values of the sequence numbers at the two ends differ by the traffic currently 'in the pipeline'. Flow control is established by the convention that the sender can only send up to N packets ahead of the last packet which was acknowledged. Equivalently, a source may not have more than N packets 'in the Pipeline' at one time. To keep the repeater code as simple as possible, N should be equal to one (packet-at-a-time) for repeater control connections. This protocol is a simplified subset of the protocol developed by Cerf and Kahn [Ref. 7] for internetwork communications.

The station maintains control over such parameters as the maximum number of packets which may be in the pipeline, and the duration between retransmissions of unacknowledged packets. Such control is used to avoid or alleviate congestion in the packet radio network.

The protocols of the Packet Radio Network are layered, or hierarchical. The program which deals with control information at level M passes control and data for all levels greater than M as transparent data. Conversely, the program which deals with control at level M does not see control information at levels less than M; it is inserted by lower level programs on transmission, and stripped off by lower level programs on reception.

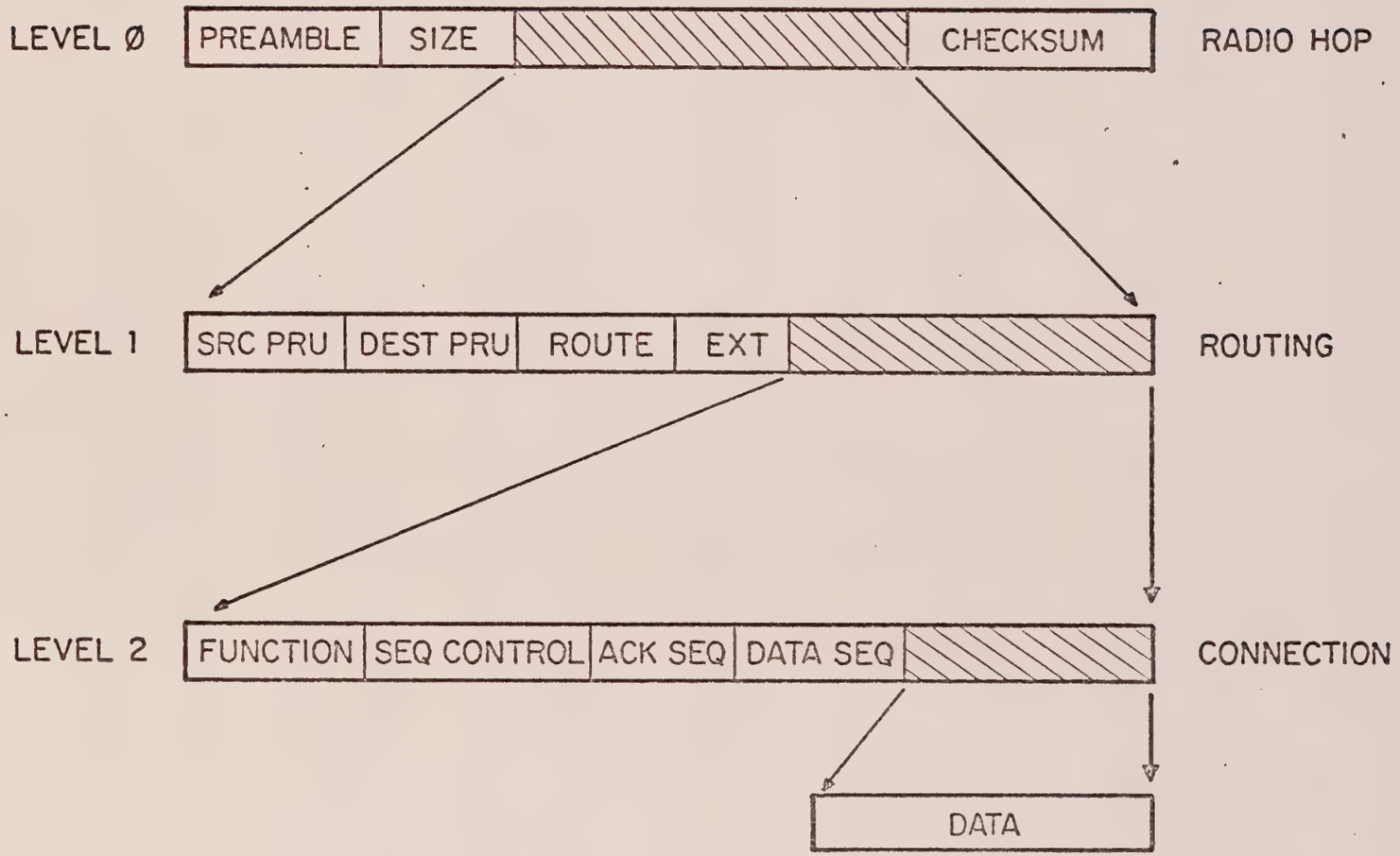


FIGURE 3: PROTOCOL FORMAT FOR SUPPORT OF A PRN CONNECTION

Figure 3 shows this layering explicitly: the connection protocol described above is the level 2 protocol, based on the level 1 routing protocol which controls the PRN store-and-forward routing for the packet. The routing protocol is itself based on a level 0 'Radio Hop' protocol which provides broadcast synchronization and error detection for transmission of the packet from one PR to the next.

The sequence numbers shown were discussed previously. The 'sequence control' field is used to synchronize sequence numbers when a connection is established, and to signal termination of the connection. The 'function fields' provides an address: within a PR, it selects the control process, the debugging process, or the measurement process.

The station has a full understanding of these protocols. It utilizes that described above for most PRN communication; a stripped down version for one-directional devices (if any) as mentioned above; with directory functions it aids in establishing a connection from a PRN source to a PRN destination; and as a gateway it translates protocols if necessary for data transmission to and from another network.

The control functions performed by a station include initialization of the PRN, dynamic routing changes, and multi-station coordination. Initialization of the PRN includes the following steps:

1. Measurement of RF propagation connectivity between all stations and repeaters. This measurement data is used to construct the connectivity matrix: this is a matrix of binary values which indicate the radio units which are capable of direct communication with each other.

2. Configuring the PRN by loading each repeater with routing parameters which control the packet store-and-forward program. These parameters specify forwarding of packets [in the direction of minimum delay] to the next repeater within "earshot".

3. Establishing control, debugging, and measurement connections from the station to each repeater that it controls. These connections remain open to perform the indicated functions as long as the station and repeater continue to function normally.

Once the station has labelled all PR's and established connections to them, the information for maintaining these connections is entered into the station's connection table. This contains status information for handling the connection protocol. As terminals come "on-line" within the PRN, each terminal is also given a connection to its controlling station, and this information

is added to the station's connection table.

Dynamic routing changes are performed locally within the PRN by permitting a repeater to specify an alternate address for the next hop after some number of unsuccessful attempts to forward the packet along its specified route. This capability provides a localized terminal tracking facility which will hand off a mobile terminal from one repeater to another. The station recognizes such operations and updates its status information for the terminal involved.

A less dynamic but more global routing change is required when a repeater or station fails: persistent alternate routing of packets signals this failure, and the station detecting it must reconfigure the PRN to route all traffic around the failed element. The station performs this reconfiguration by updating the connectivity matrix to indicate that the failed element is incommunicado; by recomputing minimum distance routes to the elements which are still active; and finally, by updating repeater routing parameters to route packets along these new routes.

The third station control function is multi-station coordination. This will be supported by PRN connections between stations and a station-station protocol. In some cases, when both stations are connected to a point-to-point network, it may be more efficient to use this network for station coordination rather than the PRN. This coordination is required in two cases:

1. Hand off of a mobile terminal as it passes out of the area controlled by one station and into the area controlled by another station.

2. An alternate station assuming all communications responsibilities for a station which fails.

These applications will require a distributed data base stored redundantly at a number of stations. The data will give status information on each active connection, and will be updated often enough that little information will be lost in the event of a station failure. Some of the PRN terminals may be accessing host resources in the ARPANET: an ARPANET reconnection protocol will be defined to permit switching the ARPANET portion of the terminal connection from the failed station to the backup station. These complexities are not being addressed in the initial experimental implementation, but they will be vital reliability measures in an operational packet radio network. Many of these issues of reliability achieved through maintenance of redundant distributed data bases have been explored in the RSEXEC (Resource Sharing Executive) distributed computation testbed. [Ref. 10]

A second phase of the experimental program being conducted at the Packet Radio testbed network at Palo Alto will demonstrate the concepts of redirecting traffic from a failed station to an

operational station, and the backup of resource allocation information.

IV. Station Debugging and Statistics Functions

A level-3 debugging protocol has been defined which supports debugging of remote PR's from terminals attached to a central station. The debugging functions include examining and depositing words into the PR memory, and setting "mousetraps" which send an error code to the controlling station when some anomalous condition occurs, e.g. hardware failure. These functions make it possible to do centralized software maintenance of remote, unattended repeaters. The maintenance terminal of the station will normally be attended by an operator or system programmer.

A similar mechanism permits centralized collection of traffic statistics, both through examination of counters in PR memory and through centralized reception of special status conditions such as "trace packets" moving through the network. Again, it was essential to centralize this function for remote, unattended repeaters.

Once the station has collected a set of traffic statistics, it will normally forward these measurements to a service host for detailed statistical analysis, logging and plotting.

V. Station Support of Nonencrypted Terminals

Some PRN applications will require secure or private communications, i.e. end-to-end encryption between a terminal and the service host which it is accessing. Assuming that the station is merely an intermediate node in the path from terminal to host, the station must be completely transparent to the scrambled data. (Any modification of the data could make it impossible to unscramble.)

On the other hand, some PRN applications do not require encrypted terminals: for these applications, the station can take over part of the terminal service function and simplify the terminals. One example of this is the TELNET RCTE option. [Ref. 9] TELNET is the protocol which translates a variety of different physical terminals into a single standard logical terminal called a network virtual terminal: this conversion may involve both code conversion and interpretation of format effector characters (tab, carriage return, etc.). This concept provides an extremely valuable simplification because previously N*M conversion routines were required to interface N terminals to M systems. Using the network virtual terminal concept, only N+M conversion routines are required.

The RCTE option of TELNET is Remote Controlled Transmission and Echoing. It permits character echoing for a full duplex connection to be performed local to the terminal (eliminating annoying echoing delays) under control of the remote service host, which may, for example, suppress echoing of typed-in passwords.

The TELNET RCTE protocol option may be too complex to incorporate into the simplest unencrypted PRN terminal. In this case, the station can handle the protocol conversation with the remote server on the terminal's behalf. Of course, the station cannot perform this service for encrypted terminals.

The TELNET process in the station may also be used to specify and set up connections to remote server hosts via a gateway connection.

VI. Internetworking Applications

So far, we have focussed on the attachment of terminals to a PRN. It is also reasonable to attach service host computers to a PRN, particularly when the network may be deployed in the absence of other communications networks, (e.g. for fleet communications). Such a host attaches to the PRN as a multiplexed set of the standard PRN connections described in Section II.

When some other network is present, it is important to provide connections between the terminals and hosts of the PRN and the terminals and hosts of the other network. This is being done for the ARPANET in two ways:

1. For communication with ARPANET hosts which support a protocol congruent with the PRN connection protocol (the Cerf-Kahn protocol mentioned previously qualifies here), the station functions as an extremely simple gateway: arriving packets are simply forwarded into the other network after their header format is converted to that of the destination network. In this case, the station does not detect missing or duplicate packets, does not reorder packets which arrive out of order: it is merely a packet reformatting and readdressing service.

2. The second approach we will use to provide station internet gateway facilities will be protocol conversion between the host-host protocols of the two networks. In particular, one connection will be established from a PRN device to a station using the PRN connection protocol described in Section II. Another (full-duplex) connection will be established from the station to an ARPANET host using the current ARPANET host-host protocol. Data arriving from either of these connections will be forwarded through the other connection.

The first approach has four obvious advantages beyond its simplicity: first, error control is truly end-to-end instead of two path (PRN device-to-station, station-to-ARPANET device). Failure of the station will not cause data loss with this approach though it would with the two-path approach.

The second advantage of this approach is dynamic rerouting after a station failure. Since the station is merely doing packet readdressing, any other station attached to the PRN and the ARPANET can do this function equally well: no connection status information is kept in the station. Recovery and continued operation of a connection after the station fails merely requires redirecting the packet traffic to another gateway station.

The third advantage of the "simple gateway" approach is that the gateway does not introduce additional delay into the end-to-end path by forcing reassembly and reordering of packets at an

intermediate location, (the station).

The fourth advantage of the "simple gateway" approach is that it supports end-to-end encryption of all data except the address headers: this can provide security against data disclosure, but no security against traffic pattern and volume analysis. The second gateway approach would require a secure station to decrypt and re-encrypt data flowing through it.

On the other hand, the second approach enjoys two advantages: first, there is independent flow control on the two connections, so 'acks' for PRN packets are returned from the station (about 200 msec) rather than from the remote host (about 1 sec). This faster turn-around of acknowledgements permits shorter source timeouts and more rapid recovery from lost packets.

The second advantage of this approach is that it permits the utilization of completely different protocols which are currently operational within the two networks, as long as the two protocols are functionally equivalent and a mapping exists between the two. This applies not only to host-host protocol, but also higher level protocols such as TELNET, file transfer, and remote job entry. All that is required is a conversion program for the high-level protocol of interest, interposed between the PRN connection and the ARPANET connection.

The experiments planned for the packet radio testbed system will explore the tradeoffs and performance advantages of these two different gateway concepts in detail.

VII. Station Structure

Figure 4 shows the hardware organization of our prototype station: it is a PDP-11 processor interfaced to a packet radio unit. In the initial tests, it will also be gateway connected to the ARPANET. The hardware interface between the PDP-11 and the PR consists of a pair of memory channels on each end, permitting full duplex DMA packet transfers.

The station requires the following hardware components:

-- PDP-11/40 central processor with:
    -- 64 K words core memory with parity
    -- KE11-E extended instruction set
    -- KT11-D memory management
    -- KW11-P programmable clock
-- IMP11-A APRANET interface (2, one to PR, one to ARPANET)
-- Packet Radio Unit
-- terminal, preferably hard copy and at least 300 baud

The PDP-11/40 was selected because of ready availability of the machine, peripherals, parts and service, and a suitable operating system (ELF; see below). Manufacturer interest in marketing the interface to be used, the IMP11-A, is a bonus in employing the PDP-11. This interface provides a standard means of accessing both the ARPA network and the station's PR, minimizing the complexity and debugging of both hardware and software.

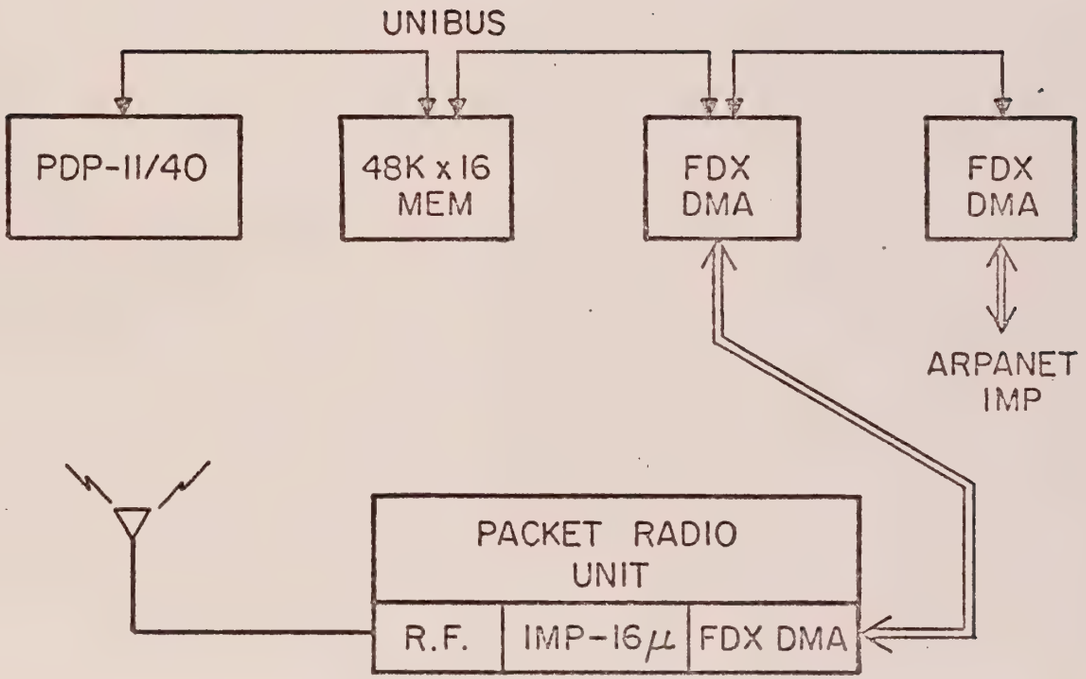


FIGURE 4: STATION HARDWARE

The software organization inside the PDP-11 is shown in Figure 5. We selected the ELF operating system* as the basic environment. This is a time-sharing operating system written in PDP-11 assembly language. ELF was selected because of the good performance of its predecessor, ELF-1; because of explicit inclusion of ARPA network message handling routines; because of its ability to support several processes simultaneously, and virtual memory spaces for these processes; and because of its availability to the ARPA community.

Processes running under ELF communicate with the ELF monitor via EMT instructions to activate input/output, to synchronize computation and data transfers, to obtain system parameters, and to manipulate processes. For a complete description of these monitor calls and their arguments, refer to [Ref. 11].

Two modifications were required to ELF to support the packet radio application: first, the PR was added as a bidirectional ELF device. This required addition of an interrupt service routine for the PR interface hardware, and send and receive subroutines within the ELF I-O multiplexor module. These routines were coded in assembly language and integrated into the ELF operating system. The second change required to ELF was the 'Network Virtual Terminal' support package which permits PRN terminals to appear identical to local ELF terminals (attached to the PDP-11) by use of a connection to the PRN terminal, a protocol-handling PRN TELNET process, and ELF's interprocess communication facility. These routines were also coded in PDP-11 assembly language and integrated into ELF. The use of internetwork protocol [Ref. 7] will permit PRN terminals to access services of hosts on other networks.

- - - - -

*Developed by the Speech Communication Research Lab., Santa Barbara, California.

KEY:
PRN = Packet Radio Network
ETE = End-to-End
I-O = Input/Output
NVT = Network Virtual Terminal
IPP = Interprocess Port
PRU = Packet Radio Unit

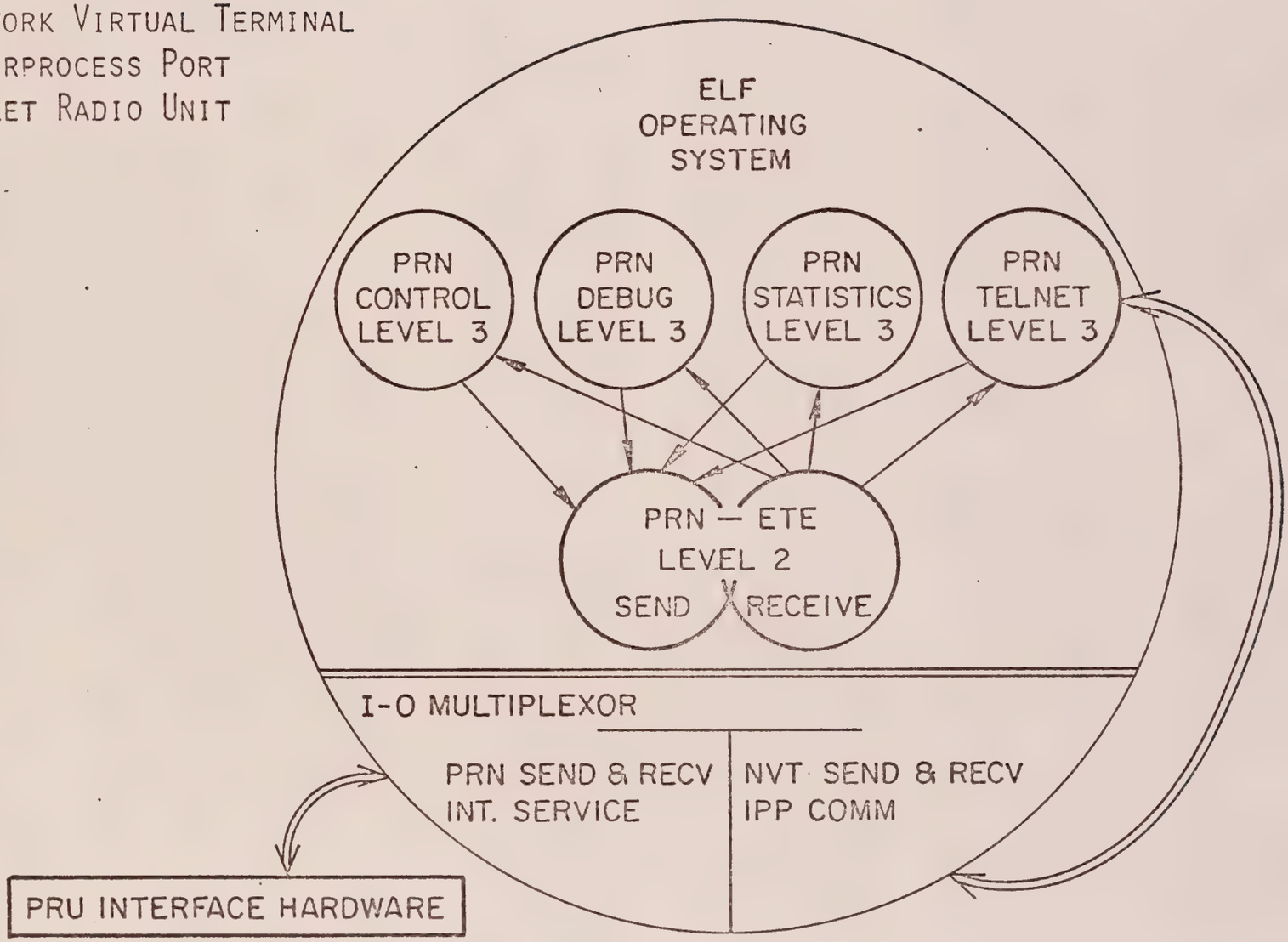


Figure 5: Station Software

An extension of the PDP-11 BCPL library routines allows direct access from user programs written in BCPL to all ELF monitor calls. This is in addition to the standard PDP-11 input and output routines.

It is possible to do the remaining packet radio software as independent user applications processes which execute under ELF. Accordingly, they are programmed in BCPL (Basic Compatible Programming Language) to obtain all the clarity, debugging, maintenance, and extensibility advantages of high-level programming. These modules are not part of ELF, and they are not integrated into the operating system.

First, a pair of processes was created which send and receive packets to and from the PR device previously integrated into ELF. The send process and receive process share a common address space which contains the connection table: this holds all status information for every connection being maintained by the station. These two processes are responsible for handling the PRN connection protocol described in Section II.

Within the same program module is a subroutine for creating a connection: this subroutine takes the address of the destination PRN device, consults the radio propagation connectivity matrix, and constructs the route which should prefix all packets sent to the specified destination. This route is stored as part of the connection status in the connection table when the connection is established.

The control process is another independent process for initializing the network and causing dynamic routing changes in response to changes in repeater-terminal propagation connectivity or repeater failures. It is responsible for keeping the propagation connectivity matrix up-to-date. This matrix is shared with the connection initialization routines which find a route to the requested device.

In the PRN initialization procedure, the Control process calls on the connection module to establish a control connection to the station's PR. It sends commands over this connection to trigger connectivity measurements (exploratory packets which request answerback from stations and repeaters within earshot). As measurement information comes back on this connection, the Control process fills in entries in the connectivity matrix, and establishes control connections to the PR's of the newly-discovered devices. This procedure is iterated until every station and repeater in the area has been configured into the network. At this point, the Control process has an open control connection to every other station and repeater in the PRN.

When any repeater detects a significant routing event, e.g. failure of some previously established route or a request from a terminal to enter the network, the repeater forwards this

information over its control connection to the nearest station. When a terminal comes on-line, the station establishes a data connection to it and provides an 'information service' to assist in completing the connection to the destination device, which may be either in the PRN or in the ARPANET. When repeater connectivity changes the station will update the connectivity matrix and reconfigure the network to bypass the failed link by modifying the routing control parameters in the affected repeaters.

The Debug program is another independent process. On request from the maintenance terminal, it calls on the connection module to open a debugging connection to the PR of interest. The debugger sends commands over this connection to examine or deposit words in the PR's microprocessor memory, and the PR responds with a positive acknowledgement for each command. There are also commands for setting traps on anomalous program conditions. When one of these conditions is encountered (assuming the PR is still operational) it sends the appropriate trap code over the debugging connection to the debugger. This types out on the controlling terminal, which is presumably attended by a system programmer or operator.

The statistics collection module is another independent process which gathers data both by examining PR memory, and by receiving statistics trap conditions spontaneously emitted by PR's. This operation parallels the operation of the debugger described above.

Finally, the PRN TELNET process performs the second type of gateway function described above: conversion between the PRN connection protocol and the ARPANET host-host protocol. Terminals on the PRN appear identical to the terminals attached to the PDP-11, and are able to access remote ARPANET service hosts in the same way.

While it would be desirable to provide additional user processes in the station (message services for PRN users, for example), our experience indicates that the memory and especially the processor time limitations preclude any processes not essential to station functions.

The station may support as many as 100 repeaters and 100 terminals. The memory requirements increase as the square of the number of repeaters if a full connectivity matrix is maintained. Station memory devoted to terminals increases only linearly, but is expected to have a larger constant of proportionality due to parameter storage of terminal characteristics and connection information. These limits are tentative and subject to revision as new features are desired, tradeoffs made to service more terminals, etc. We expect to never have to fragment a message (except for gateway functions), since the PRN message size (maximum about 142 words) is easily accomodated in all phases of processing in the PDP-1. Reliable simulations of service times are not yet available from our co-contractors, but we expect that station response will not be the predominant factor in response timing.

VIII. Conclusion

Successful demonstration of the packet radio concepts will lead to new digital communications services for mobile terminals which are reliable, difficult to detect or jam, and which make efficient use of the electromagnetic spectrum by providing coexistence with current uses of the spectrum. Possible applications include both command and control communications and secure digital voice.

The system structure described above will provide reliable, efficient and maintainable support for packet radio network communications.

IX. References

[1.] The Organization of Computer Resources into a Packet Radio Network; R.E. Kahn.

[2.] ALOHA Packet Broadcasting, a retrospect; N.Abramson, R. Binder, F.Kuo, Okinaka, Wax

[3.] A Technology for Packet Radio; Garrett, S.Fralick

[4.] Random Access Techniques for Packet Radio Networks; L.Kleinrock, F.Tobagi

[5.] Packet Radio Network Design, System Considerations; H.Frank, R.VanSlyke,I.Gitman

[6.] Digital Portable Terminals; S.Fralick, D.Brandin, F.Kuo, Harrison

[7.] A Protocol for Packet Network Intercommunication; V.Cerf, R.Kahn

[8.] Function Oriented Protocols for the ARPA Computer Network; S.Crocker, J.Heafner, R.Metcalfe, J.Postel

[9.] Remote Controlled Transmission & Echoing TELNET Option; D.Crocker, J.Postel. RFC #581.

[10.] A Resource Sharing Executive for the ARPANET; R.H. Thomas.

[11.] ELF System Progammer's Guide, D. L. Retz, J. R. Miller, J. L. McClurg and B. W. Schafer

[12.] ELF Kernel Programmer's Guide, D. L. Retz, J. R. Miller, J. L. McClurg and B. W. Schafer

[13.] BCPL Manual, BBN, September 1974